Design of Tree Architecture of Fuzzy Controller based on Genetic Optimization

Chang-Wook Han*, Se-Jin Oh**

Abstract

As the number of input and fuzzy set of a fuzzy system increase, the size of the rule base increases exponentially and becomes unmanageable (curse of dimensionality). In this paper, tree architectures of fuzzy controller (TAFC) is proposed to overcome the curse of dimensionality problem occurring in the design of fuzzy controller. TAFC is constructed with the aid of AND and OR fuzzy neurons. TAFC can guarantee reduced size of rule base with reasonable performance. For the development of TAFC, genetic algorithm constructs the binary tree structure by optimally selecting the nodes and leaves, and then random signal-based learning further refines the binary connections (two-step optimization). An inverted pendulum system is considered to verify the effectiveness of the proposed method by simulation.

Keywords : fuzzy controller, fuzzy neurons, genetic algorithm

I. Introduction

Fuzzy logic controller has been attracted great attention from both the academic and industrial communities. Fuzzy logic controller allows for a simpler, more human approach to control design and do not demand the mathematical modeling knowledge of more conventional control design methods. As systems become more complex, the ability to describe them mathematically becomes more difficult. For this reason, fuzzy logic controller provides reasonable, effective alternatives to classical or state-space controllers [1]-[5].

Rule number reduction is important for fuzzy control of complex processes with high dimensionality. As the number of input and fuzzy set of a fuzzy system increase, the size of the rule base increases exponentially and becomes unmanageable (curse of dimensionality) [6]. To solve this problem, tree architectures of fuzzy controller (TAFC), which consists of AND and OR fuzzy neurons, are proposed in this paper. The structure of proposed TAFC is different from that of conventional logic-based fuzzy neural networks to reduce the number of fuzzy rules effectively. TAFC has flexible tree structure by optimally placing fuzzy neurons as a function and selecting relevant input sub-spaces as leaves. The fuzzy neurons exhibit learning abilities as they come with a collection of adjustable connection weights [6]. In this setting of fuzzy neurons, the synergy of learning and transparency is well articulated. In the development stage of TAFC, we use two-step optimization where genetic algorithm (GA) develops the binary tree structure by optimally selecting the nodes and leaves, and then random signal-based learning (RSL) [7] further refines the binary connections. An inverted pendulum system is considered to show the validity of the proposed TAFC.

II. Structure of TAFC

Before proceeding with the detailed architecture of TAFC and learning realized for the over all network, we will briefly remind the two basic types of logic-based neurons as being introduced in [6]. AND neuron aggregates input signals (membership values) $X = [x_1\ x_2 \cdots x_n]$ by first combining them individually with the adjustable connections (weights) $W = [w_1\ w_2 \cdots w_n] \in [0,\ 1]^n$ and afterwards globally ANDing these results,

$$y = \mathrm{AND}(\mathbf{x}\ ;\ \mathbf{w}) = T_{i=1}^{n}(w_i\ s\ x_i) \qquad (1)$$

where t- and s-norms, i.e. T and s, are used to

* Dept. of Electrical Engineering, Dong-Eui University
** Radio Astronomy Division, Korea Astronomy and Space Science Institute
투고 일자 : 2010. 6. 16 수정 일자 : 2010. 7. 29
게재확정일자 : 2010. 7. 29
※ This work was supported by Dong-Eui University Grant(2008AA164).

represent the AND and OR operation, respectively. The structure of OR neuron is dual to that reported for AND neuron, namely,

$$y = \mathrm{OR}(\mathbf{x};\mathbf{w}) = S_{i=1}^{n}(w_i \ t \ x_i) \tag{2}$$

The AND and OR neurons realize pure logic operations on the membership values. Some obvious observations hold. (i) For binary inputs and connections, the neurons transform to standard AND and OR gates. (ii) The connections close to zero (one) identify the relevant inputs in the AND (OR) neuron. (iii) The parametric flexibility is an important feature to be exploited in the design of the networks. In all experiments, we consider these triangular norms and co-norms to be a product operation (a t b=ab) and probabilistic sum (a s b=a+b-ab), respectively.

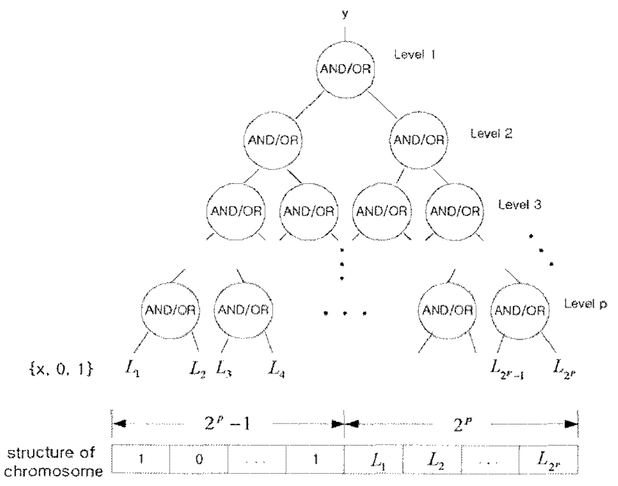


Fig.164. Structure of TAFC.

Fig. 1 shows the structure of TAFC using logic-based fuzzy neurons being viewed here as a generic means of forming the skeleton of the logic model. In this structure, each node (AND/OR neuron) and leaf (L) can select one of fuzzy neurons (AND/OR) and input sub-spaces, respectively. Obviously, TAFC has flexible structure by allowing {0,1} in every leaf to enhance the performance, i.e. eliminate useless connections from TAFC, and can express any logic by selecting proper "Level".

III. Optimization of TAFC

A. Genetic Algorithm

A genetic algorithm (GA) [8][9] is a search algorithm based on an analogy with the process of natural selection and evolutionary genetics. It starts from a set of random strings, called individuals of population, and proceeds repeatedly from generation to generation through genetic operators. The structure of the chromosome (individual) to construct a Boolean structure of TAFC is shown in Fig. 1.

A conventional simple GA has three basic operators: reproduction, crossover, and mutation. Reproduction probability is proportional to the fitness value of a string. In the process, the fitness is calculated for each individual by using the fitness function as follows:

$$F = \frac{1}{1+Q} \tag{3}$$

where Q is the performance index. Crossover implies the mating of two individuals. The information of two randomly selected individuals is partly interchanged at the crossover site. Crossover is applied to elicit valuable information from the parents, and is applied with a crossover probability. The mutation operator insures against bit loss and can be a source of new bits, or diversity. Since mutation is random through the search space, it must be used sparingly.

B. Random Signal-based Learning

Random signal-based learning (RSL) [7] is a kind of reinforcement learning that is expressed in the following form:

$$weight(t+1) = weight(t) + \eta r(t) f(n(t)) \tag{4}$$

where η is the learning rate, f() is the activation function as shown in (6), n(t) is the discrete random process with the values in [-1,1], and r(t) is the reinforcement signal which is defined as follows:

$$r(t) = \begin{cases} 1 & \text{if } \Delta Q < 0 \\ 0 & \text{if } \Delta Q \geq 0 \end{cases} \tag{5}$$

where ΔQ is the change of the performance index which will be defined later. In this learning law, synapses learn only when the performance index decreases after learning (r(t)=1). Otherwise, the learning is rejected (r(t)=0). The activation function f() is a bipolar sigmoid function:

$$f(x) = \frac{2}{1+e^{-\lambda x}} - 1 \tag{6}$$

The main idea of RSL is that the random process n(t) randomly agitates the state in the range of learning rate in order to find the optimal state. RSL is very effective to find the local optimum because the candidate solution moves in a downhill direction very quickly.

C. Two-step Optimization of TAFC

To battle the problem of exponential increase of the rule, GA attempts to construct a Boolean structure of TAFC by selecting inputs, including {0, 1}, as leaves and fuzzy neurons as nodes that shape up the tree architecture, and then concentrate on the detailed optimization of the connections (weights) connected to each nodes by RSL. RSL is a kind of reinforcement learning algorithm that is very effective to find the local optimum because the candidate solution moves in a downhill direction very quickly. During GA optimization, the connections to AND and OR neuron set as zero and one, respectively, because of the characteristic of the fuzzy neurons as mentioned before. RSL refinement involves transforming binary connections into the weights in the unit interval. RSL refinement considers only the tree connections, but the eliminated connections, which occur by the leaves with the value zero or one, are not considered as shown in Fig. 2. This enhancement aims at further reduction in the value of the performance index.

IV. Experimental Results

To show the performance of the proposed method, TAFC is applied to balancing an inverted pendulum on a cart. Let $x_1=\theta$ (angle of the pole with respect to the vertical axis) and $x_2=\dot{\theta}$ (angular velocity of the pole), then the state equation can be expressed as follows [7]:

$$\dot{x}_1=x_2 \qquad (7)$$
$$\dot{x}_2=\frac{(M+m)g\sin x_1-(F+mlx_2^2\sin x_1)\cos x_1}{(4/3(M+m)-m(\cos x_1)^2)l}$$

where M (mass of cart) is 1.0Kg, m (mass of pole) is 0.1Kg, l (half length of pole) is 0.5m, g (gravity acceleration) is 9.8m/s2, and F is the applied force in Newton. In this simulation, the following conditions are considered: initial states $(\theta[rad], \dot{\theta}[rad/s])$ are (0.3, 0) and (-0.3, 0); boundary conditions for θ, $\dot{\theta}$, and F are [-0.5,0.5], [-1,1], and [-25,25]; population size=50; generation number=100; crossover rate=0.9; mutation rate=0.03; learning rate in RSL=0.01; iteration number in RSL=500; time step=0.01s; simulating number of time step q=200; and the following performance index (fitness) that has to be minimized is used

$$Q=\sum_{i=1}^{q}(\theta_i^2+\dot{\theta}_i^2) \qquad (8)$$

Because our focal point is TAFC and its two-step optimization, we assume that fuzzy sets of the input and output interfaces are given in advance as 3-uniformly distributed triangular membership functions with an overlap of 0.5 and left unchanged [6]. For more complex rule bases, 5-uniformly distributed triangular membership functions are considered too. For the defuzzification, center of area (COA) method is used. To reduce the redundant information, the number of input to each node is fixed as two that can be increased for high dimensional problem, and only the number of Level (NL) is varying between 2 and 4 to find reasonable NL for this example. Table 1 and 2 reveal the averaged best performance index over ten independent simulations. For the GA, standard version, including tournament selection and multi-point crossover, is used.

Table 1. Averaged best performance and no. of rule for 10 trials (NFS=3).

	NL=2	NL=3	NL=4
After GA	12.31	10.99	10.98
After RSL	10.18	9.90	9.89
No. of rule	5.3	9.0	9.0

Table 2. Averaged best performance and no. of rule for 10 trials (NFS=5).

	NL=2	NL=3	NL=4
After GA	10.96	10.17	10.15
After RSL	9.87	9.60	9.58
No. of rule	9.4	23.6	25.0

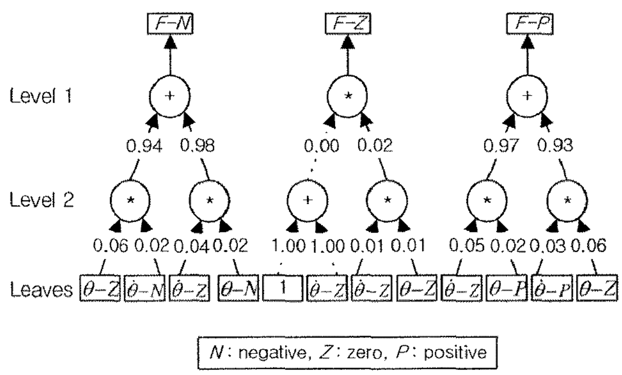


Fig.2. Constructed TAFC (NFS=3, NL=2).

Fig. 2 shows one of the constructed TAFC for the number of fuzzy set (NFS)=3 and NL=2. In this figure, the connections expressed as dotted line mean the eliminated connections by GA and are not considered for RSL refinement. The other connection weights are further refined by RSL as already described in Table 1 and 2. In Fig. 2, "*" and "+" of the nodes represent AND and OR neuron, respectively. If we assume that all inputs in the leaves are binary, we can read the rule as follows:

"if(θ-Z AND $\dot{\theta}$-N) OR (θ-N AND $\dot{\theta}$-Z) then F-N"

"if(θ-Z AND $\dot{\theta}$-Z) then F-Z"

"if(θ-P AND $\dot{\theta}$-Z) OR (θ-Z AND $\dot{\theta}$-P) then F-P"

From this TAFC, we can take the five rules without overlapped (redundant) rule, while the maximum rules (3^2) that can include overlapped rule are generated for NL=3 and 4 as shown in Table 1 and 2. Though overlapped rule can be included in the maximum rules, there is no conflict in the rule base because GA globally optimizes the binary structure of TAFC by decreasing (8).

Table 3. Averaged results for testing initial conditions (NFS=3).

	NL=2	NL=3	NL=4
Performance	8.86	8.59	8.58
No. of failure	0.0	0.0	0.0

Table 4. Averaged results for testing initial conditions (NFS=5).

	NL=2	NL=3	NL=4
Performance	8.53	8.27	8.21
No. of failure	0.0	0.0	0.0

To check the validity of the resulting TAFCs, simulations for the initial states, (-0.3,-0.3), (-0.3,0.0), (-0.3,0.3), (0.0,-0.3), (0.0,0.3), (0.3,-0.3), (0.3,0.0), and (0.3,0.3), are carried out. As a result of this simulation, the averaged performance and the number of failure (counting the number exceed the boundary conditions) are illustrated in Table 3 and 4.

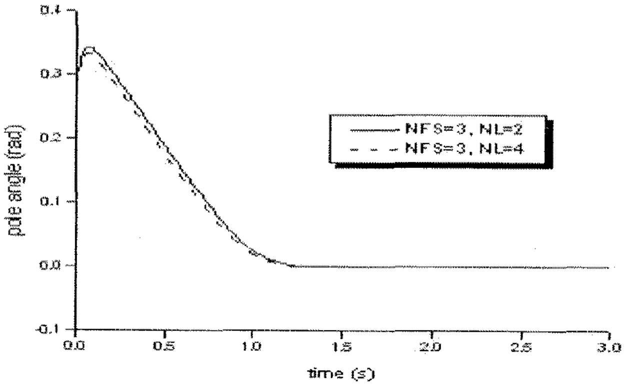


Fig.3. Averaged simulation results for NFS=3 with the initial condition (θ, $\dot{\theta}$)=(0.3, 0.3)

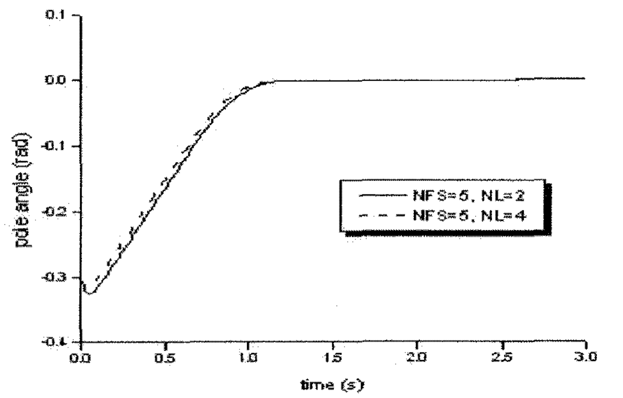


Fig.4. Averaged simulation results for NFS=5 with the initial condition (θ, $\dot{\theta}$)=(-0.3, -0.3)

Fig. 3 and 4 describe the averaged simulation results using the optimized TAFCs with NL=2 (reduced rules) and 4 (maximum rules) for each NFS, and initial conditions are (0.3, 0.3) and (-0.3, -0.3) for NFS=3 and 5, respectively. Obviously, the performance of NL=4 is better than that of NL=2 for each NFS as described in Table 1 and 2, but the simulation results in Fig. 3 and 4 indicate that the reduced rules are enough for balancing the pole.

V. Conclusions

TAFC that has flexible tree structure as well as the learning and interpreting ability by using fuzzy neurons has been demonstrated. For the development of TAFC, two-step optimization, where GA develops binary structure and then RSL further refines the binary connections, has been considered. From the simulation results, we believe that the proposed TAFC effectively reduce the number of rules with reasonable performances by selecting proper NL.

References

[1] L. X. Wang, "Stable adaptive fuzzy controller with application to inverted pendulum tracking," *IEEE Trans. Systems, Man, and Cybernetics-PartB*, vol. 26, no. 5, pp. 677-691, 1996.

[2] G. H. Hwang, H. J. Hwang, D. W. Kim, and J. H. Park, "A fuzzy logic controller for speed control of a DC series motor using an adaptive evolutionary computation," *Trans. Control, Automation, and Systems Engineering*, vol. 2, no. 1, pp. 13-18, Mar. 2000.

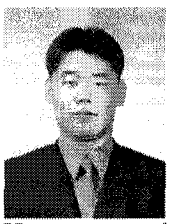
[3] C. C. Hua, Q. G. Wang, and X. P. Guan, "Adaptive fuzzy output-feedback controller design for nonlinear time-delay systems with unknown control direction," *IEEE Trans. Systems, Man, and Cybernetics-PartB*, vol. 39, no. 2, pp. 363-374, April 2009.

[4] T. P. Zhang, H. Wen, and Q. Zhu, "Adaptive fuzzy

control of nonlinear systems in pure feedback form based on input-to-state stability," *IEEE Trans. Fuzzy Systems*, vol. 18, no. 1, pp. 80-93, Feb. 2010.
[5] H. Ohtake, K. Tanaka, and H. O. Wang, "Fuzzy model-based servo and model following control for nonlinear systems," *IEEE Trans. Systems, Man, and Cybernetics-PartB*, vol. 39, no. 6, pp. 1634-1639, Dec. 2009.
[6] W. Pedrycz, M. Reformat, and C. W. Han, "Cascade architectures of fuzzy neural networks," *Fuzzy Optimization and Decision Making*, vol. 3, no. 1, pp. 5-37, 2004.
[7] C. W. Han and J. I. Park, "A study on hybrid random signal-based learning and its applications," *International Journal of Systems Science*, vol. 35, no. 4, pp. 243-253, 2004.
[8] J. H. Holland, *Adaptation in natural and artificial systems*, Ann Arbor, MI: University of Michigan, 1975.
[9] D. E. Goldberg, *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley, Reading, MA, 1989.

Chang-Wook Han

received the B.S., M.S., and Ph.D. degrees in Electronic Engineering from Yeungnam University in 1994, 1996, and 2002, respectively.
He joined the faculty of the Department of Electrical Engineering, Dong-Eui University in 2008, where he has been an assistant professor. His research interests include fuzzy-neural networks, intelligent control, computational intelligence and its applications.



Se-Jin Oh

received the B.S., M.S. and Ph.D. degrees in Electronic Engineering from Yeungnam University in 1996, 1998, and 2002, respectively.
He was employed by the Department of Information and Communication, Taegu Science College as a full-time lecturer from 2001 to 2002. In 2002, he joined the faculty of the Radio Astronomy Division, Korea Astronomy and Space Science Institute, where he is now a senior research scientist. His current research fields are digital signal processing for speech recognition, VLBI correlator, radio astronomy related equipments and its applications.